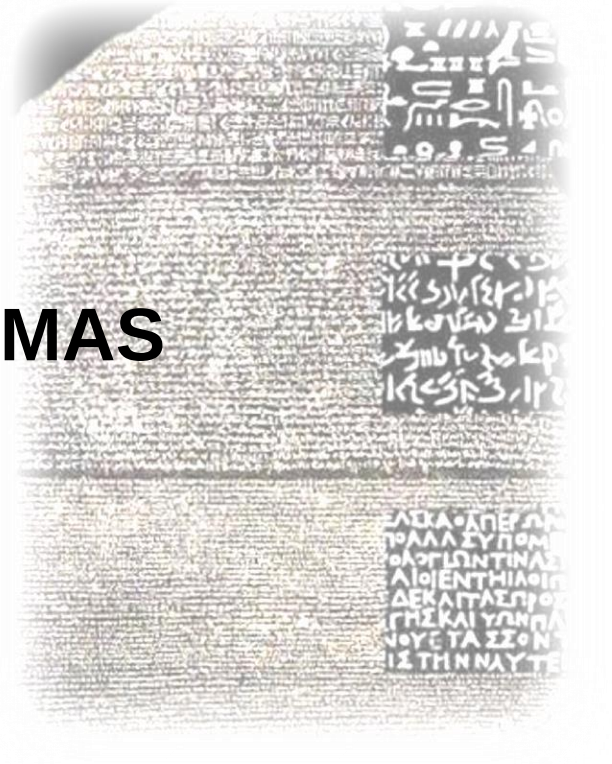
ITPA Databases
Mapping to IDS for archiving in IMAS



Olivier Hoenen
ITER Organization

Disclaimer: The views and opinions expressed herein do not necessarily reflect those of the ITER Organization

Motivation

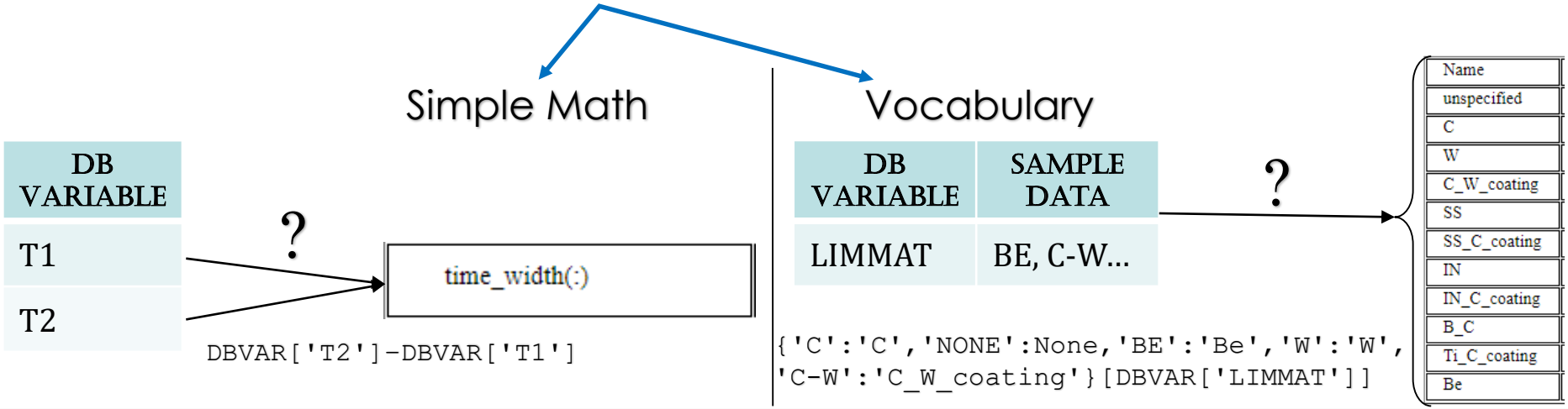
- Data archiving
  - Avoid loss of valuable legacy DB (e.g. L-mode)
- Improve exploitability of the data
  - Unique definition (IMAS Data Model) of quantities across the different databases
  - Standardization of storage and access
  - Possible extension of database entries with more complex data
- Improve definitions of some IDS of the IMAS Data Model

Getting started

- Small project of Marc Vidal
  - Currently high school student (US)
  - 1 month internship at IO in July 2021
  - Draft mappings between Global H-Mode Confinement DB and IDS
    - Recent update, extensive description of variables at https://osf.io/hrqcf/
  - Aim at developing simple tool to support physics expert to express their mappings in a generic manner
    - Mapping files in plain text
    - Might be reused for other DBs that share same definitions
  - Allow automation with minimal IMAS knowledge
  - Final report on IDM: 5J5JKR

Current mapping

- Generic mapping file is being defined for 0D data
  - Simple text file in `csv` format, contains 3 relevant columns
    - The source variable name which is being mapped from the original DB
    - The path within the target IDS which represents the same quantity
    - An optional transformation expression



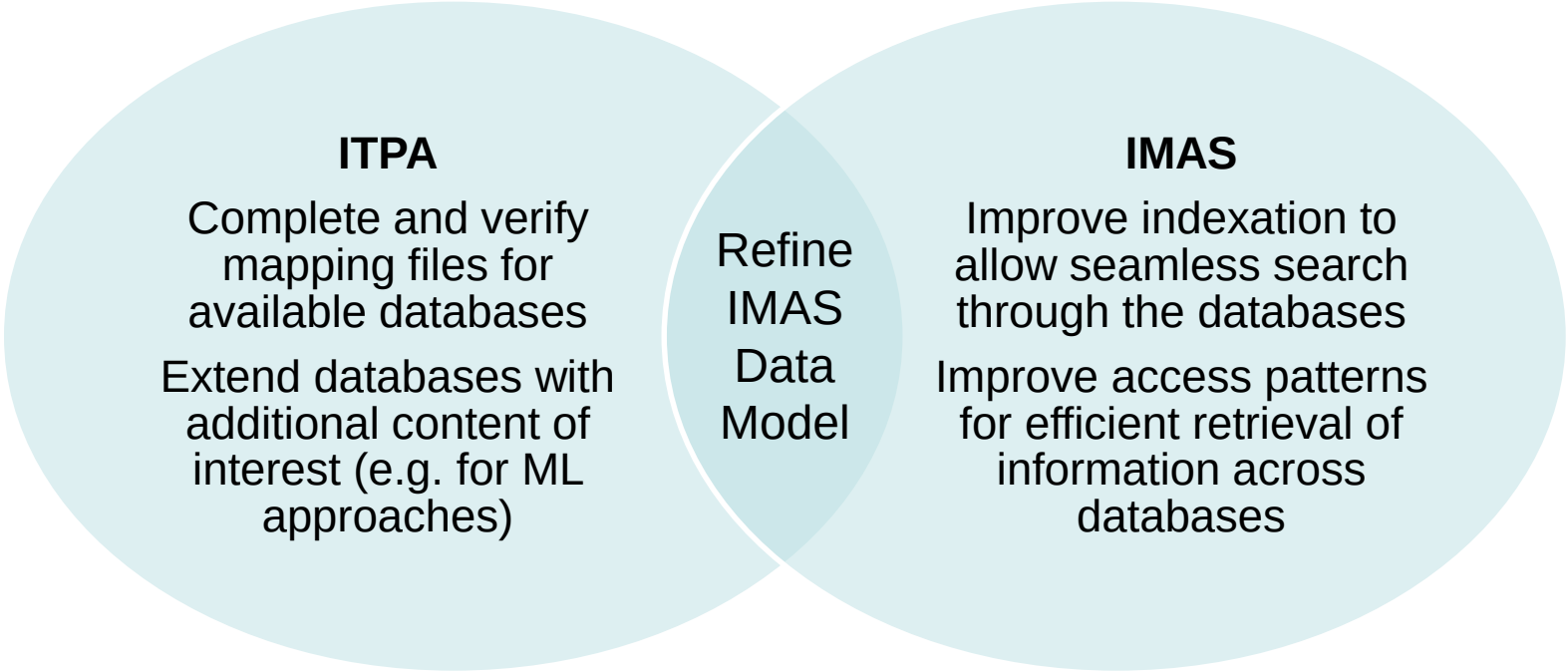
Current mapping (2)

- Chosen conventions
  - IDS path starts with IDS name, sub-structures are ‘/’ separated e.g. `summary/global_quantities/ip/value`
  - Transformations
    - A valid Python expression (will be evaluated)
    - All DB variables for current row will be accessible through a DBVAR dict, to allow combining several DB variables in one expression e.g. `2*(DBVAR['BEILI2']-DBVAR['BEIMHD'])`
    - Vocabulary transformation is written with evaluated dictionaries e.g. `{'C':'C','NONE':None,'BE':'Be','W':'W','C-W':'C_W_coating'}[DBVAR['LIMMAT']]`
- H-Mode mapping is 45% complete (among 191 variables)

Mapping tool

- Simple Python script
  - Expect the external database to be loaded as Pandas dataframe
    - Can understand various formats with minimal effort, including csv
    - Allows powerful manipulations and access patterns
  - Take a mapping file (for a chosen external DB) as argument
  - Can store IDS using the various IMAS Access Layer backends (MDS+, HDF5, plain text)
  - Each DB entry (line) is stored as a separate IMAS DB entry
- Available in the IDStools repository (first release: 1.10.0)

Open points and next steps



For getting in touch: imasusers on Slack
For questions or additions to IMAS Data Model: JIRA